

09/865,802 PTO-892

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
16 February 2006 (16.02.2006)

PCT

(10) International Publication Number
WO 2006/016350 A2

(51) International Patent Classification: Not classified

(21) International Application Number:
PCT/IL2005/000693

(22) International Filing Date: 29 June 2005 (29.06.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
163448 10 August 2004 (10.08.2004) IL

(71) Applicant (for all designated States except US): TEL
HAI ACADEMIC COLLEGE [IL/IL]; 12210 M.P.
Upper Galilee (IL).

(72) Inventors; and

(75) Inventors/Applicants (for US only): YULI, Itzhak
[IL/IL]; 60 Tel Hai Ave., 11031 Kiryat Shmona (IL).
PIECHOWICZ, Hagay [IL/IL]; Kibbutz Elrom, 12466
M.P. North Golan Heights (IL).

(74) Agent: MILLER - SIERADZKI, ADVOCATES &
PATENT ATTORNEYS; P.O.B. 6145, 31061 Haifa (IL).

(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN,
CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI,
GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE,
KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA,
MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ,
OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL,
SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC,
VN, YU, ZA, ZM, ZW.

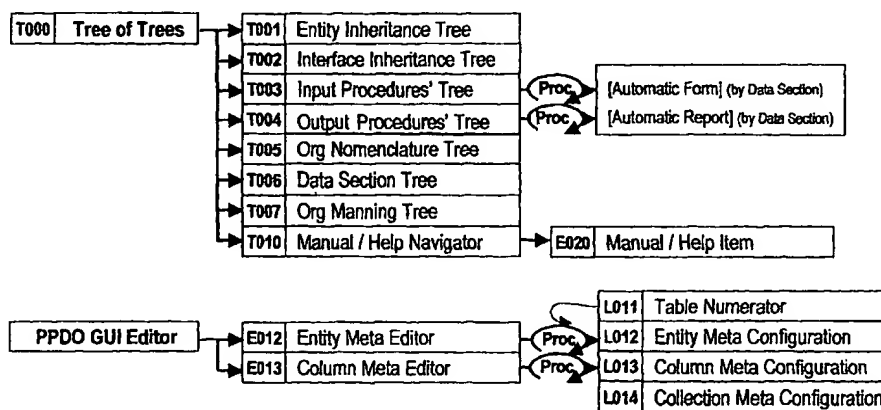
(84) Designated States (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, GH,
GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,
ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,
FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO,
SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN,
GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished
upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guid-
ance Notes on Codes and Abbreviations" appearing at the begin-
ning of each regular issue of the PCT Gazette.

(54) Title: ENTERPRISE RESOURCE PLANNING BASED ON SEPARATION OF PERCEPTION AND DATA



Legend

Tree Perception Tree Entity Table Collection Inheritance Pointer Macro

(57) Abstract: A method for a universal Enterprise Resource Planning (ERP) information management, suitable for any organiza-
tion. The information is separated into raw data and at least one of a plurality of subjective perceptions attached to it. The method
comprises: organizing information into tables of three types: entity table for raw data, tree table for perception, collection table for
many-to-many associations, and manipulating the information according to a set of predetermined parameters that constitutes meta
characteristics of the information, using a set of macros to perform operations.

WO 2006/016350 A2

ENTERPRISE RESOURCE PLANNING BASED ON SEPARATION OF PERCEPTION AND DATA

FIELD OF THE INVENTION

- 5 The present invention relates to enterprise resource planning. More particularly, it relates to a new approach to databases, introducing a novel concept for databases, and method of their use thereof.

BACKGROUND OF THE INVENTION

- 10 Managerial information is commonly handled by enterprise resource planning (ERP) software packages. These are database-driven tools that are heavily customized for the specific use of a given customer, or at least for a highly defined customer discipline. As a result, most of the essence of the customer's managerial information pattern is embedded at the code level. Since this pattern is the result of 'soft' algorithms,
15 and since the number of managerial solutions for a given problem is more than one, organizations tend to be characterized by a rather complex model with a relative large number of degrees of freedom.

- Traditionally, developing ERP applications at the code level to accommodate the customer pattern generally results in intricate and highly expensive solutions.
20 Furthermore, these solutions tend in most cases to represent the perception of the programmers (usually these programmers are not members of the customer's organization) who are tailoring the ERP platform for the customer, rather than the perception of the customer's own information technology (IT) manager. In any event, the tailoring is usually so specific that any subsequent unanticipated evolution of the
25 customer organization requires reopening of the code, something most programmers consider impractical. In conclusion, as much as an organization requires an ERP tool for achieving better performance, the organization is dependent on external expert tailoring of the ERP tool and is compelled to freeze development of the tool at the point where the tailoring was completed.

- 30 Furthermore, ERP systems are not geared to support human mistakes, therefore

they cannot be used as a conceptual aid for a learning manager. Thus, the traditional ERP hinders the development of the modern idea of a learning organization.

To achieve an improved ERP, it is necessary to reevaluate the nature of managerial information, which is what ERPs are designed to help manage. An analysis
5 of managerial information reveals a complex mixture of two modes of information: raw data and perceptions (or concepts) applied to data. While the common practice is to ignore the differences between the two modes of information, it is our opinion that the undistinguished approach to this mixture in ERP tools is the underlying cause of the tools' limitations in both the technical and the managerial sense. One should appreciate
10 that raw data is kept pretty much constant along the lifespan of the organization, managerial perceptions, on the other hand, reveal considerable variation with changes of personnel and with the development of the organization over time. Hence, these two modes of information, raw data and perceptions, are of an utterly diverse nature. In fact, they actually constitute two independent dimensions of the information body, within
15 which they co-exist. By contrast, traditional ERP storage of data in flat tables is a reflection of a rudimentary one-dimensional approach to information processing wherein other features are handled at the code level. Therefore, while ERP systems provide good and reliable means of raw data manipulation at the customer level, system modification to accommodate changes in the customer's perception of how his business
20 is organized must be done at the programmer level – in other words, beyond the reach of the ordinary organizational personnel, such as the most natural candidate for the job, the organization's IT manager.

Applying human characteristics to mechanistic constraints is widely approached in the artificial intelligence work done in association with efforts to render robots with
25 signal perception and environment perception. Two disciplines are largely considered in approaching these goals. These are fuzzy-logic, which is meant to challenge marginal uncertainty, and neural networking that is aimed at rendering computing devices a grasp of systematization.

While others have already pointed out that information includes perceptual
30 components, this notion has not trickled into the realm of databases, particularly to ERP products. We, offer, for the first time, a mechanistic definition and real apparatus that enable the automation of human perceptual features with regard to information. It is

3

achieved by defining a minimal set of database features, which are sufficient for the determination of the perception component of managerial information. These features support:

The hierarchical and nested tree arrangement of perceptual items.

5 Automatic preservation of, and accessibility to, past states of the perceptual trees.

Unlimited sequential association between any information item, and its element.

Rendering each information item sufficient links that systemization is conceivable from any relevant point of the information body.

10 *Ad Hoc* intra-organizational association-disassociation of the trees.

Attenuation of linguistic ambiguity, while maintaining multi-linguistic capability, including the establishment of organizational nomenclature.

15 It is a system that is readily capable of adaptation to conceptual alteration, be it a result of human mistakes or the fruits of a learning procedure. As such, it becomes automatically an instrument for depicting the past states of the organization. Hence, it provides the user with an ever growing sense of self, as well as a record of organizational experience. Other individuals may follow the evolution of the organization by accessing the perceptual trees at various previous points in time.

20 The dynamic nature of the hierarchical trees not only registers the user's perception, it can be easily used as an organizing aid in the construction of a new concept.

25 Organizations may gain, as well as share, perceptual knowledge with peer and/or supervisory organizations by association and disassociation of their relevant conceptual trees. "Knowledge" is considered by the inventors as the contexture of properly associated perceptions, and is the third layer in the hierarchal series where "raw data" is the lowest, followed by "perception", "knowledge" and finally rises to the degree of "intelligence".

In this way governmental and regional organization may gather and distribute information and data with affiliated organizations. In cases where a large body of

4

perceptual information is needed for the proper functioning of an organization, (e.g., in the fields of education, health care, legal precedents, etc.), tree sharing enables automatic joint endeavors of knowledge construction. This joint product is concomitantly available to all tree sharing organizations.

5 Accordingly, several objectives and advantages of the present invention are:

 An ERP method and system that enables the automation of human perceptual features with regard to information.

 An ERP method and system that is programmable at the user-level, requiring only a familiarity with a rather limited vocabulary.

10 An ERP method and system that provides view of the organization's knowledge at any point in its history.

BRIEF DESCRIPTION OF THE INVENTION

15 There is thus provided, in accordance with some preferred embodiments of the present invention, a method for a universal Enterprise Resource Planning (ERP) information management suitable for any organization, the information separated into raw data and at least one of a plurality of subjective perceptions attached to it, the method comprising:

20 organizing information into tables of three types:

 entity table for raw data,

 tree table for perception ,

 collection table for many-to-many associations,

 manipulating the information according to a set of predetermined parameters that
25 constitutes meta characteristics of the information, using a set of macros to perform operations.

 Furthermore, in accordance with some preferred embodiments of the present invention, the tables comprise a database.

5

Furthermore, in accordance with some preferred embodiments of the present invention, entity table comprise administrative fields, which record inheritance.

5 Furthermore, in accordance with some preferred embodiments of the present invention, entity table comprise administrative fields which record reciprocal pointing relations.

Furthermore, in accordance with some preferred embodiments of the present invention, entity table comprise administrative fields which record interface implementations.

10 Furthermore, in accordance with some preferred embodiments of the present invention, entity table comprise administrative fields which record affiliation to an echelon of the organization.

Furthermore, in accordance with some preferred embodiments of the present invention, entity table comprise informative fields for raw data.

15 Furthermore, in accordance with some preferred embodiments of the present invention, tree-table comprise records defining tree-nodes.

Furthermore, in accordance with some preferred embodiments of the present invention, tree-table comprise multi-lingual fields.

20 Furthermore, in accordance with some preferred embodiments of the present invention, one of the multi-lingual fields relates to a nomenclature of the organization.

Furthermore, in accordance with some preferred embodiments of the present invention, tree-nodes comprise fields relating to tracking of evolution of the tree-nodes.

25 Furthermore, in accordance with some preferred embodiments of the present invention, tree-nodes comprise fields relating to compartmentalization.

Furthermore, in accordance with some preferred embodiments of the present invention, tree-nodes comprise fields relating to one or many

entity or collection records.

Furthermore, in accordance with some preferred embodiments of the present invention, tree-nodes comprise fields relating to tree-node to tree-node association and/or disassociation.

5 Furthermore, in accordance with some preferred embodiments of the present invention, tree-nodes comprise fields relating to a cascade of tree-nodes and thereafter relating to their related tree-nodes and/or entity records and/or collection records.

10 Furthermore, in accordance with some preferred embodiments of the present invention, collection table comprises administrative fields holding one-to-one relations between records.

Furthermore, in accordance with some preferred embodiments of the present invention, collection table comprise administrative fields holding one-to one relation between a record and a value.

15 Furthermore, in accordance with some preferred embodiments of the present invention, collection table comprise administrative fields which record reciprocal pointing relations.

20 Furthermore, in accordance with some preferred embodiments of the present invention, the set of predetermined parameters comprises one or more tables.

Furthermore, in accordance with some preferred embodiments of the present invention, the tables comprise a database.

Furthermore, in accordance with some preferred embodiments of the present invention, the set of predetermined parameters is modifiable.

25 Furthermore, in accordance with some preferred embodiments of the present invention, the set of predetermined parameters that constitutes meta characteristics is stored in a kernel.

Furthermore, in accordance with some preferred embodiments of the present invention, the set of predetermined parameters that

constitutes meta characteristics of entities.

Furthermore, in accordance with some preferred embodiments of the present invention, the set of predetermined parameters that constitutes meta characteristics of fields.

5 Furthermore, in accordance with some preferred embodiments of the present invention, the set of predetermined parameters that constitutes meta characteristics of collections.

Furthermore, in accordance with some preferred embodiments of the present invention, the operations also include functions.

10 Furthermore, in accordance with some preferred embodiments of the present invention, the operations include automatic operators that create or modify tables and further meta characteristics using a predetermined editor.

15 Furthermore, in accordance with some preferred embodiments of the present invention, the operations dynamically create automatic input forms and/or output reports in accordance with the meta characteristics.

Furthermore, in accordance with some preferred embodiments of the present invention, the automatic input forms include a contracting/expanding tree.

20 Finally, in accordance with some preferred embodiments of the present invention, the operations dynamically guard validity of data in accordance with the meta-characteristics.

25

BRIEF DESCRIPTION OF THE FIGURES

The invention is described herein, by way of example only, with reference to the accompanying Figures, in which like components are designated by like reference numerals.

FIG. 1 illustrates a core of an ERP application system based on separation of perception and data, in accordance with a preferred embodiment of the present invention.

FIG. 2 illustrates Fixed Columns, in an exemplary ERP application, in accordance with a preferred embodiment of the present invention..

FIG. 3 illustrates Meta-Definitions of tables, in an exemplary ERP application, in accordance with a preferred embodiment of the present invention.

FIG. 4 illustrates Meta-Definitions of columns, in an exemplary ERP application, in accordance with a preferred embodiment of the present invention.

FIG. 5 illustrates a module of an ERP application in accordance with a preferred embodiment of the present invention that was designed for Tel-Hai College, showing several perceptual approaches to the same set of data (in this case space management).

FIG. 6 illustrates a core of an ERP application system based on separation of perception and data, in accordance with a preferred embodiment of the present invention, suitable for adaptation for any organization, with the addition of compartmentalization core.

FIG. 7 illustrates a module of an ERP application in accordance with a preferred embodiment of the present invention that was designed for Tel-Hai College marketing department.

DETAILED DESCRIPTION OF THE INVENTION

The present invention is directed to a data processing system and method for use in enterprise resource planning (ERP).

The present invention is based on the realization that information, such as information about an enterprise's resources and management, has two main characteristics: impartial raw data and how that raw data is subjectively perceived by a

particular individual or organization.

As an example consider an educational organization such as a college. The college administration has to allocate classrooms and other types of building spaces for classes and other activities. A classroom may be defined as a walled space, but for a teacher not every walled space can serve for classes (for example storage rooms, hallways, etc.), however for the maintenance staff, all walled spaces must receive the same upkeep, and for them a "classroom" has the same meaning as a "storage room". In other organizations same term may be assigned an utterly different meaning.

In other words, same raw data has different perceptions attached to it by different organizations. Moreover, same raw data (like "walled space") has different perceptions attached to it by different people of the same organization.

A main aspect of the present invention, in view of the above realization, is the novel separate treatment of the different aspects of information. The present invention deals with the arrangement of rational database by way of structure and function in a manner that takes into account the different aspects of raw data and its variety of corresponding perceptions, and based on that database arrangement a novel Enterprise Resource Planning (ERP) application (sometimes referred to, in this specification, as PerDB) is introduced.

This perceptual database serves as the basis of a generic ERP application. Whereas usually an ERP application is specially tailored by programmers to a specific client's needs, the ERP application of the present invention is provided as a general framework, that each client can adapt himself to his organization's unique and dynamic needs. Hence, the balance of resource requirements is shifted from programming to characterization. In this way, the present invention promotes leadership in management since the emphasis in the customization phase should focus primarily on appropriate mechanization of the organization's structures, procedures and rules (in-house or external technical help from IT professionals may be needed to set up a customized system).

The perceptual database of the present invention can be implemented in many ways, as will be apparent to one skilled in the art. In a preferred embodiment, it is implemented in at least three sets of tables (tree, entity, collections –explanation

hereinafter).

In accordance with a preferred embodiment of the present invention, a set of general principles may be summarized as follows:

- 5 (1) Information is comprised of objective Raw-Data coupled to Subjective Perception(s).
- (2) Data may be coupled to an unlimited number of Perception(s).
- (3) Data is approached (*i.e.*, reading or writing) only via a Perception.
- (4) Data is acquired only by predetermined Input-Procedures (see Input Procedures Tree T002, FIG. 1) by Automatic Forms.
- 10 (5) Since Data is coupled to Input Processes, which are in turn triggered by known Events, Data and Events are tightly associated, to the extent that recorded data illustrates unequivocally its generated events.
- (6) Analysis of Data is subject to predetermined, yet adaptable, Output Procedures (see Output Procedures Tree T003, FIG. 1) by automatic forms.
- 15 (7) Perception is subjective (pertaining to an individual, or a group) and evolvable by learning.
- (8) The basic topology of a perception is depicted by the Perceptual-Tree table, comprising of an Infinite nested generations of Tree-Nodes (*i.e.*, records of the Perceptual-Tree table).
- 20 (9) Perceptual-Trees specificity character implies that the properties of a Tree-Node trickle down to its children tree-nodes, till new properties are attributed to a down stream Tree-Node, which start a new set of properties trickling to its own children tree-nodes.
- (10) Data is stored in Instances, which are records of Entity tables.
- 25 (11) A complex Instance may be composed of various Instances that maintain a certain inheritance relation among themselves as defined by the Entity-Inheritance-Tree (see Person Card E025 and Personal Items E026 tables, FIG. 6).
- (12). Perceptual-Trees and Entities may intralink by pointing one at either itself,

or interlink by pointing at each other. Single-to-plural pointing is feasible by Collection tables comprising of Joint records.

(13) Each Instance item holds knowledge of all relevant data items (Tree-Node, Instance or Joint) that are pointing at itself. Thus, all relevant information, pertaining to a given Instance, may be reconstituted from any point of the data body.

(14) Cross effects on different field values may be achieved by Macros embedded in the field's meta-features.

(15) Accessibility is denied by default to everyone. It is open only to predetermined members (actors) of the organization.

(16) A user is granted accessibility to information by specific (see Tree specificity) matching of three orthogonal attributes (DataSec, short for data section, *i.e.*, a reference to the nature of the data, SubOrg, short for sub-organizational unit, *i.e.*, a reference to an affiliation to an organizational echelon, and Action Privileges) with the two corresponding orthogonal DataSec and SubOrg flags that are carried by each information item.

The enterprise resource planning application based on separation of perception and data is conceived according to a mechanistic model wherein understanding of knowledge of an enterprise is implemented at three levels of information management:

- (A) definition and storage of raw data
- (B) perceptual approach to data
- (C) information-networking as the basis of knowledge

Raw data is defined as data items comprising details at the utmost required/relevant resolution. The arrangement of these data items must maintain at all times independence of each item from every other item, from its usage, and/or its interpretation.

Raw data has to be stored in such a way that it will be globally retrievable. Information Security is achievable by a two-phase matching mechanism. On the first hand, each data element (*i.e.*, Entity field or Tree-Node) is *a priori* imprinted by the two orthogonal dimension flags: DataSec and SubOrg. By that a data element carries one

attribute to its data-nature, as by the organizational perception displayed in Data Section Tree T006 (see FIG 1), and a second attribute to its affiliation to the appropriate organizational echelon, as by the Org Manning Tree T006 (see FIG 1). The other phase of the matching mechanism requires that retrieving agents should carry proper accessibility keys, i.e., the three dimensions: DataSec, SubOrg and Action Privileges, the latter should be appropriate for the adequate handling of the desired data. In other words in order to be allowed to carry out a certain action on a specific data, a user must possess full matching of DataSec and SubOrg flags with the corresponding flags associated with that specific data as well as Action Privilege that allows him to carry out that action.

Most people cannot relate to an impartial data but by assigning it a verbal name. Most people will refer, in any practical conversation, to a plane of wood, about one square meter in area, attached to four legs of 75 cm in length as a "table". However, "table" is merely a common layman's interpretation. That is so, since on one hand, no craftsman is capable of reconstructing a "table" unless real materials and measures are specified, and on the other hand, the same structure can be put to use as a base for a statue, as a spacer, as a door barricade, or various other functions. Yet, no matter how this item would be referred to, it will always comprise raw data: a certain plane of certain matter with certain four legs.

Verbal naming has been historically the most fundamental and unavoidable first level perceptual attribute to data. Yet, one should be aware that naming is naturally subjected to a certain ambiguity, the degree of which is largely dependent of the speaking-language used. Furthermore, it is empirical experience that a given body of raw data may be perceived differently by different people of different nationalities. On top of that, information communication between humans is frequently complicated by differences in linguistic interpretations of parallel verbal themes. In short, perceptual usage methods must accommodate cross-cultural management.

It should be appreciated that unlike computerized retrieval engines, which may locate a data item directly at its register, the human approach to information, raw data included, is carried out first via its perceptual tag, generally, by its verbal name. Moreover, perceptions of a given body of (unvarying) raw data are most probably changing continuously as knowledge advances.

The average human psychophysical capability of simultaneously handling a very limited number of independent pieces of information, forces us to think with the aid of grouping mechanisms (e.g., stereotyping, generalization, encapsulation, etc.) to which verbal tagging is usually assigned. With time the groups themselves are considered as information.

Perceptual confrontation with complex systems made up of a large number of items is achieved by means of categorization. It's a method of hierarchically tagging items with common denominators into groups that are themselves gathered into supra-groups based on meta-common denominators, and so forth. It has been employed as a taxonomical systematization method in almost every scientific discipline, and is the basis for database navigational engines. Hence, hierarchical trees are one leading concept of the perception component of information.

Coping with the perceptual dimension of information is accomplished mechanistically by perception-tree structures which maintain four intrinsic structural degrees of freedom (apart from the organizational parameters, such as security):

- (A) Reduction of verbal ambiguity to a minimum.
- (B) Maintaining a flexible hierarchical nested tree arrangement.
- (C) Tracking the evolution of the perception with time.
- (D) Tracking the ownership of the perceptual tree,

Linking inherently independent facts (raw data) and perception trees to a common metaphysical superstructure, usually referred to as the "whole picture", constitutes an individual's comprehension, whereas further association of information into larger superstructures constitute knowledge. The ability to render information structures the capability of being adequately inter-associated (i.e., linking) is the mechanical basis for transforming data and perceptions to a sophisticated body of knowledge that should aid organizational personnel in building comprehensive system perceptions, as well as aiding them with the tough job of decision taking.

The potential for interlinking information is embedded in all components of the present invention. Raw data tables may include pointer fields which register an expansion of data. It may be a single data item of an orthogonal (i.e., to the table's entry

dimension) nature, or a collection of data elements. A data field may point at a perceptual tree element whenever it refers to a grouped meaning of variant information perception, which in turn may point at any other data or collections of data.

5 Perceptual tree items are obviously expected to point at raw data, for which they serve as perceptual entry points for human beings. However, the same perceptual items may associate themselves with other perceptual items that are mounted on either the same or completely separate perceptual trees. These are the means for endowing a system with multi-dimensional aspects. For instance, space management at Tel-Hai College involves the perceptual construction of campus, buildings and rooms (see table 10 "walled space tree" T100 in FIG 5). From a different perspective (maintenance dept.) a different perceptual tree is constructed (see table "space maintenance tree" T601 in FIG 5), pointing at the same set of data entities. Yet another perspective (this time for the purpose of allocating spaces with accessibility to handicapped persons) may involve the construction of a third perceptual tree (see table "affiliated levels" T600 in FIG 5), that 15 points at branches of the original perceptual tree (T100), and through its nodes, pointing at the same set of data entities.

the therapeutic effect of a drug, which by itself is a member of a medication tree, is associated with the disease tree, with the HMO financial tree, and with the malpractice hazard tree by a non-linear series of pointers. Certainly the inter-linking of these trees is 20 a crucial feature in conceiving the body of knowledge necessary for a properly informed real-world medical decision.

Finally, while inter-organizational sharing of information is a key element in maximizing knowledge within the organization, it can at times detract from healthy competition or the organization's self-esteem. Accordingly, while the present invention 25 enables peer organizations to seamlessly associate their perceptual trees, it also enables them to disassociate the linkage at will. Intra-organizational accessibility of the perceptual trees is unaffected by inter-organizational association or disassociation.

The inventors of the present invention suggest defining a so-called "perceptual database" (or PerDB for short), separating data from perception, and providing an ERP 30 application that is in fact a basic framework into which any organization may pour its contents, environment and concepts, thereby being equipped with all necessities for

building its own ERP application without the help of external programmers. Technically PerDB serves as an interpreter of objects (i.e., a series of specifically defined tables, which will be elucidated thereafter) and engines whose data is indifferently embedded in any ordinary relational database.

5 PerDB utilizes the database merely for data storage and retrieval, and does not define or use database diagrams, table relationships or other high-level performance tools. Relational diagrams and links between data entities (tables) that are commonly imprinted in the particular code of specific applications are, in the present invention, dealt with by PerDB interpretation of meta-data (stored in specific, yet, ordinary entity
10 tables) according to object oriented guidelines as explained hereinafter. By doing so, the present invention becomes independent of the specific database platform (examples of platforms include: MS-SQL, Oracle, DB2 and others).

"Inheritance", "polymorphism", "encapsulation" and "Interface implementation" are considered as fundamental characteristics of object oriented programming, whereas
15 they are completely absent from conventional relational databases. In the present invention, however, these characteristics are introduced into PerDB.

Inheritance is a way of expanding the definition of a previously defined base entity to create a new entity, which defines a "is-a" relation between the base entity and the inheriting entity, e.g., a rocking-chair is a chair, therefore inherits properties common
20 to all chairs. Inheritance allows reuse of code.

In the past (we refer to relational database programming), the definition of the rocking-chair table had to include the definition of the chair-table and any other definitions of higher base tables.

In the present invention, PerDB implements the "is-a" relation based on meta-
25 level definitions. Consequently, the data elements of the base tables needs not to be repeated for reuse.

Polymorphism is a way of relating to a group of varying entities by their common properties, as these are defined in their common base entity, e.g., a rocking chair, a wheel chair and a stool are all chairs.

30 In the past, each variant of the chairs was represented by a separate table, which

had to be explicitly referenced. Furthermore, if more than one of the varying entities is to be pointed at, the referencing entity had to include all options of the available varying entities in its definition. In addition, when changing the inheritance scheme (adding or removing a variant entity under the common base entity) all referencing entities had to be modified to systematically reflect the change.

In the present invention, since meta-level already includes all relations of a referencing entity with all other entities with which it has defined inheritance relations, the referencing entity needs to define only a single reference to the appropriate common base entity. Consequently, when referring to a certain base entity all its inheriting variants are all acceptable. In addition, this also implies that when making changes in the inheritance scheme no additional modifications of the referring entities is needed.

Encapsulation defines behavior of an entity without the need of an external authority (i.e. each entity contains all of its data, functions and properties).

In the past the programmer had to write a specific code for validation rules, value limits, default values, and other properties for every form that referenced an entity, thus the quality of the product was largely dependent on the quality of the programmer.

In the present invention, all PerDB-objects (in accordance with object oriented programming) encapsulate their own features, thus systematically reflecting these features in every form.

An Interface is a predefined set of features. When entities are related to interfaces they each define a "can" and/or "has" relation. For example, some chairs are stackable and therefore "can" be stacked. When implementing an interface (e.g., "stackable") in different entities (e.g. chairs, plates, boxes), it is possible to synchronically treat the implementing entities in a polymorphic manner in spite of their otherwise apparent mutual exclusiveness..

The basic framework of the present invention consists of defined building blocks (i.e., PerDB-objects) with predetermined characteristics and behavior patterns, and their organization on top of it (through its IT professionals or other professional help) can set up an ERP application (end-application), to be used by the end-users – them being administrators, workers and clients alike, according to the specific organization's needs. This approach allows each organization to take this basic framework and harness it to its

managerial needs and purposes, instead of turning to highly skilled, yet out of the organization, programmers for a tailor-made solution. This also means that maintenance work is no longer the task of external specialists, who often make their tailor-made applications too complicated to be maintained (let alone understood) by others. Instead
5 the organization can assign its own workers to maintain the system, and fix problems, as these occur. Note, that problems may occur, as the end-application is designed by people for use by people. The inventors of the present invention stipulate that after a reasonable learning period, the basic framework will become a powerful, yet easy-to-use tool, which would render the organization managers firm-yet-flexible opportunities to
10 exercise leadership in a learning organization that is free of external restrictions imposed by tailor-made solutions.

The ERP application based on separation of perception from data is based on three primary concepts. In the preferred implementation, these are PerDB-objects implemented as database tables comprising records. It will be recognized by one skilled
15 in the art that they could equally be implemented in other programming formats, for example as objects in a high level programming language.

The three PerDB-object tables are:

- (A) Tree tables constructed of Tree-Node records.
- (B) Entity tables constructed of Instance records
- 20 (C) Collection tables constructed of Joint records

As mentioned, perception in the present invention is taken to comprise four degrees of freedom. To implement these degrees of freedom Tree-Nodes comprise fixed definitions (see the tree Fixed Columns, FIG 2) assembled in four pseudo-blocks. There
25 is also a fifth pseudo-block that serves as an operational component.

In a Tree-Node the Tree-Node's verbal tagging ambiguity may be reduced to minimum with the following fields:

NodeID: A permanent reference to a perceptual item. This is a notation that will never expire as long as the organization exists. A deletion of a perceptual item is
30 never executed physically, but rather handled by its status flag, associated with the

"Since" and "Till" fields (see below).

Label_H: The common verbal notation commonly used within the framework of the local language (H stands for "home", and refers to the native language of the users – the present invention was invented in Israel, and the actual application that was built was in Hebrew).

Label_E: The English translation of the common notation of the perceptual item (E stands for "English"). Translation to English (or other reference language chosen for the system) provides a considerable reduction in the verbal ambiguity, as sometimes the same word in a certain language has several meanings. Consider, for example, the word "chair", which may be a piece of furniture but also a description of an appointment. In most cases, when switching to a second language this duality is lost, as for each of these meanings is a separate word in the other language. Though this field primarily serves to reduce verbal ambiguity, it also constitutes a pivot for multi-lingual applications.

NodeType: A nomenclature notation taken from the organization's terminology. NodeType points to a system tree which embodies all key words that had been used or are anticipated to be used by the organization (see Org Nomenclature Tree T004, FIG 1).

In cases, such as this, where a field points to a tree, the user interface can include a combination box or drop down list that displays the tree in contracting/expanding mode (sometime referred to as ComboTreeBox for short). One method for doing this is to display initially the top level nodes with an indicator whether a node has children and where the user can select the node to open the next level of the tree in the list.

In a Tree-Node, the Tree-Node's order in the Tree hierarchy is indicated by the following field:

NodeOrd: This can be a string that follows legal notation (i.e., n.n.n), wherein the numeration indicates the branch and level. For example, value 2.1.3 indicates that the Tree-Node is on branch 2, sub-branch 1, and sub-sub-branch 3.

The quantity of possible branches (sibling nodes) and generations (children nodes) in a tree is effectively limitless, and may vary according to subjective opinion of

personnel within different organizations. A Tree-Node's characteristics trickle down to successive tree-nodes (nested children). Once a down the tree-node is attributed a new characteristic that overwrites the previous value and is inherited (in the same procedure) by its own successive (down stream) tree-nodes. In this way tree-specificity is defined as the overriding of a higher generation characteristic with respect to lower generation one. Tree-specificity is restricted to within a given tree branch. Inter-branch specificity requires careful Tree-Node associations which will constitute a virtual tree structure that would produce the same specificity pattern.

A Tree-Node position on a perceptual tree is flexible. Tree owners are free to move a Tree-Node from any point to any other point on the tree arrangement. When a Tree-Node is moved elsewhere in the tree, it retains all its features except the hierarchical meanings that is derived from its new position.

In a Tree-Node, the tree-node's evolutionary qualities are treated by the following fields:

ID: The key field (i.e., DB terminology) of the Tree-Node record. Unlike NodeID, which remains unique for that Tree-Node for the lifespan of the organization, the ID identifies the Tree-Node as it pertains to a given time window. A time window is defined as the time period within which all of the Tree-Node's parameters (i.e., content of fields) remain the same. A change in any field is considered a change in the Tree-Node's status; therefore a new time window is created.

Since: The point in time (date:hour:minute:second) of the beginning of a new time window. This value is updated automatically by the system.

Till: The termination time (date:hour:minute:second) of an existing time window. This value is updated automatically by the system.

NodeStatus: The meaning of the field depends on the enterprise's organizational policy and/or style. Hence, the values of this field are taken exclusively from (i.e., pointers at) the Status branch of the organizational nomenclature tree (see Org Nomenclature Tree T004, FIG 1) the same tree used for NodeType.

Various perception trees can be constructed for the same body of data. In this way, an organization may hold to an official view (i.e., a perception tree held by an

appointed user or user-group) as well as an unlimited number of individual views (*i.e.*, private perception trees) of authorized personnel.

In a Tree-Node, the ownership of the Tree-Node is a property that is handled within the compartmentalization (*i.e.*, accessibility and security) block by the following
5 four fields:

Editor: The owner of a node can be either a certain individual or a user-group, which eventually holds a list of certain individuals. The value of this field is checked against the PerDB-login details of the user. A user who holds action privilege status of edit or higher (such as delete, insert, initiate) to a Tree-node, may become a
10 Tree-Node Editor holding exclusively all privileges derived from the tree-node's fields and data pointed at. Editor rights trickle down hierarchically the perceptual tree. Each time a new editor is specified within a tree branch, it implies that an editor is delegating editor authority of the new individual to that node, its offspring nodes and to the information entities on which they point at.

15 DataSec: This field is meant to limit accessibility to the Tree record by indicating the accordance of the nature of data contained in the Tree record with regard to a given type or essence of organizational data as specified on the Data Section Tree (T006, FIG 1). The DataSec field exerts the data-type correlation holding a pointer at a proper certain Tree-Node of the Data Section Tree. It goes without saying that this data-
20 type correlation pertains to all children Tree-Nodes of the pointed at Tree-Node (*i.e.*, sometimes related to as a tree branch) of the Data Section Tree too. It should be noticed that the data-type correlation pertains only to one particular branch of the Data Section Tree. Would there raise a need to pertain to more than one nested branch of the Data Section Tree, the IT personnel should consider the construction of another adequate
25 hierarchy of branches.

SubOrg: This field is meant to limit accessibility to the Tree record by indicating the pertinence of data contained in the Tree record with an appropriate subsection within the organizational echelon. The content of this field is a pointer at a Tree-Node of the Organization Manning Tree (T006, FIG 1). Here again, a pointing onto
30 an organizational node means that particular section and all of its subordinating ones.

Protected: A Boolean value which distinguishes between information items

deemed absolutely necessary for the proper functioning of the system, and those items which are in the responsibility realm of the organization (i.e., its IT personnel or laymen users).

5 Properties: A technical field that is used for the introduction of macro procedures that are primarily meant to exert specific functions on remote Tree, Entity and/or Collection fields.

10 The prime essence of the perceptual tree is to approach raw data by the adequately permeable perceptual parameters and filters, as depicted above. This goal is reached by the data block fields, as follows:

15 Data: If the perceptual node is aiming at a single informational item. In case that it consists of a single data (e.g., a string, a number, a time point, etc.), a quantity (i.e., numeric value coupled to a series of dimensions) or a statistical product (e.g., number of instances, sum, etc.), then this field would hold the informative data itself in the appropriate syntax.

 In the case that the node record is referring to another record (i.e., Tree-Node, Entity Instance or Collection Joint), the "Data" field will hold a string address, which PerDB interprets as a pointer onto the specified record.

20 In the case that the node record is referring to a Collection Joint, PerDB recognizes that this pointer refers actually to the assembly block of target records as by the definitions of the auxiliary Collection table (see L014, FIG1).

25 DataType: As can be appreciated from the preceding paragraph, "Data" fields may hold a series of expressions, each of different syntax and a different essence. In order that PerDB system would properly interpret the content of the "Data" field, it requires an adequate indication. This is achieved by the DataType auxiliary field, which is by itself a pointer onto the "Field" branch of the organizational nomenclature tree (T004) that may include (see FIG 2):

 Self Values, such as (but not limited to) text, memo, object, Boolean expression, time, date, time & date, time window, date window;

30 Numbers, such as (but not limited to) Integer, automatic numerator, real

number, percent, quantity;

Dummy field, which is an empty of content auxiliary item.

Statistic products, such as (but not limited to) counter, counter + sum, counter + mean value, counter + mean value + SD value (standard deviation);

- 5 Pointers, selected from the three types of pointer (see above) plus a TreeBranch pointer used for tree-tree association;

Execution button is a trigger for activating a macro procedure.

Due-Day Alert is a trigger for an alert message to be activated on the first occasion following the date and time specified.

- 10 DataSrc: This field is an auxiliary field aiming at assisting the IT personnel by limiting the span of search in a Tree table, depicted for the IT in the ComboTreeBox, to the appropriate branches of the organizational nomenclature tree within the relevant indication of the "DataType" field.

- 15 Raw data accumulates in storage tables designated "Entity", whose records are referred to as "Instance"s.

- 20 Entities are created and designed as derivatives of the Entity Heritage Tree (T001, see FIG 1), in a fashion parallel to the formation of a new child-node. It permits the user to formulate complex assemblies, diversely appending a variety of entities to a base entity, hence gaining the object-oriented quality of polymorphism of the "is a" relation. For example (see FIG 5), the Entity records of rooms/working spaces of a sophisticated organization, such as a higher education teaching institute, are efficiently organized in an object-oriented cascade framework. Under such a scheme the record of a computer classroom will be split among the base instance of "a space" table, and the extending instances "a room", "a classroom", and "a computer classroom" tables. Each
25 instance holds certain specific fields at their proper specialization level. The parallel record of an office will be split among "a space", "a room" and "an office" instances, respectively.

- 30 Entity Instances may also be coupled with complementary Interface Instances, which contain information shared with other unrelated Entities, in an object-oriented polymorphism of the "can" and/or "has" relation. For example, mailing information may

be encapsulated in certain Instances in the appropriate Interface. Then, this information can be implemented by coupling it with a variety of entity instances that have a mailing address by themselves. Beyond the efficiency of this polyphormic mechanism, over the repetitive fields in every Instance, Interfaces facilitate readily co-gathering of various Entity Instances according to the shared implemented information.

Entities (interfaces included) employ a series of control components that should take care of three types of systematic performances:

- (A) Expression of proper compartmentalization flags.
- (B) Self-orientation (object-oriented like and neural-network like) within the bulk of information.
- (C) Validation of appropriate content (i.e., type of data) and functions (i.e., pointers and macros) of the instance fields.

To comply with these systematic demands, an Entity is coupled to two series of meta-definitions (see FIGs 3 and 4) and to a series of administrative fixed columns (see FIG 2). Both meta- and administrative components are used *Ad Hoc* by PerDB interpreter for its run time procedures. These meta- and administrative definitions serve as substitutes to the commonly used code for controlling the behavior of ordinary ERP applications, thus, facilitating seamless and immediate responsiveness to regulatory modification within the authority of the organizational IT personnel.

The first level meta definitions attributed to the Entity level can be edited by the IT in the "Entity Meta Editor E012" Entity (see FIG 1). Upon approval of the meta-definitions values (as stated in E012 fields), they are transformed to control mode and transferred to the "Entity Meta List L012" Entity by an appropriate macro procedure (see FIG 1). PerDB performance is controlled according to the definitions found in L012 fields at each execution.

The Entity meta-table level comprises entry parameters made of the following components:

Entity_CN: The notation name of the Entity comprised of a prefix E or I (for Entity or Interface, respectively) followed by a unique three digit number (e.g., E020, E101, I431) drawn from the "Table Numerator L011" (see FIG 1).

Label_H: The local home-Language verbal name of the Entity.

Label_E: The English translation of the Entity Name.

The accessibility issue is taken care of at the Entity level by the following Components:

5 DataSec: A pointer at a certain Tree-Node of the Data Section Tree (T006, FIG 1) which serves as an indication that the content of the Entity table pertains to a given type or essence of organizational data in a manner similar to that of the DataSec-field effect in the Tree-Node record (see above).

10 SubOrgFieldCN: An auxiliary field that is meant to assist the IT personnel in way of providing a default value for the herein SubOrg field by drawing a certain remote SubOrg field content.

SubOrgOverride: A second auxiliary field that is meant to assist the IT personnel in way of providing a certain default value for the herein SubOrg field.

15 Entity table complies with the system consistency demands (PerDB enforceable) in accordance with the following control meta-fields:

Abstract: A Boolean indicator for the fact that the Entity cannot initiate Instances on its own. Consequently, the IT is prevented from creating/initiating any Instance to the Entity. It rather serves as an intermediate generation in the Entity heritage cascade, as by the object-oriented standards.

20 Final: A Boolean indicator for the fact that the Entity is the last of a heritage cascade (i.e., no Instance can be inherited from the Entity)

CN_i (CN stands for "common name"), i serves as a numerator used by PerDB in automatic naming of new Fields.

25 Collection_i is a parallel technical field for regulating the automatic formation of new collections.

DenyDendrites A Boolean indicator for the fact that the Entity does not require knowledge of who points at him.

Abstracts A vector (i.e., an array) of Interface tables that are implemented by the Entity.

Subsequently to the Entity meta-table parameters, which serve as control indicators for PerDB gate for entering the Entity as a whole, PerDB requires indicators for functioning at the proper resolution within the entity's rows (i.e. Instance) and columns matrix. Accordingly, additional control parameters are provided in the two
 5 formats: the Administrative Fixed Columns (see FIG 2)., which pertain to the rows' control; and Column Meta-Definitions associated with the Instance's Data Columns (see FIG 2) which pertain to the Entity's column control.

Administrative columns are meant to supports the PerDB proper systematic interpretation of Entities (Interfaces included) with regard to:

- 10 (A) Self-orientation (both object-oriented like and neuron net-work like) within the bulk of information.
- (B) Affiliation to sub-organization echelon (as first compartmentalization flag).
- (C) Implementing adequate Interfaces.

15 The orientation issue is supported by the following five fields:

ID : An identification of the Instance.

UpPtr: A pointer at the preceding instance in the object-oriented heritage cascade.

20 DownPtr: A pointer at the successive instance in the object-oriented heritage cascade.

Dendrites: A series (i.e., an array) of reciprocal pointers at all other Tree-Nodes, Instances and Joints which point at the Instance ("who points at me?"). This neural-network like parameter is meant to keep the systematic integrity of the data body. In addition, it serves as a safety tool that should prevent the deletion of an Instance that
 25 is needed for the proper functioning of other parts of the system. In should be noted that this feature renders the present invention neural-network qualities, hence permitting accessibility to a desired point in the data from any relevant entry-point within the data body.

The affiliation to the sub-organizational echelon is supported by the field:

SubOrg : A pointer at the Organization Manning Tree (T006, FIG 1) bearing identical features to those specified for Tree's and meta-Entity's SubOrg fields (see above) .

5 The implementation Interface Instances is supported by the field:

Interfaces : A sequence (*i.e.*, an array) of Interface-Instances in accordance with the Interface sequence specified in the Entity's meta-definitions (see "Interfaces" meta-field, FIG 3).

10 The information columns of the Entity may hold a spectrum of data types and syntax. It reflects the notion that PerDB supports the storage of not only plain strings and numbers, but also quantity values (*i.e.*, a numerator coupled with its dimension) and pointers at every possible Tree-Node, Instance and/or Joints. To comply with that versatility, an Entity informative column is controlled by a series of characteristics, which
15 are initially edited with the aid of the Column Meta Editor E013 (see FIG1) by the organizational IT personnel. Upon approval the parameters are transformed into control mode and transferred by an appropriate macro procedure to the L013-Field Meta List (see FIG 1).

20 The series of the Entity meta-Column parameters are assembled in four pseudo group of configuration fields.

The first group of fields serves to identify the name of the data column by the following characteristics:

Table: A reference to the Entity or Interface table name in accordance with the table specified on the Entity Meta List L012 (see FIG 3).

25 CN: The column "Common Name" of the field comprised of a "CN_" prefix and a numeric index derived from the "CN_i" meta-field of the Entity Meta List L012 (see FIG 3). The common name is the column reference used by PerDB interpreter.

Label_H: The field's humane alias of the column in the local home

language.

Label_E: The field's humane alias in English.

The second group of fields comprises meta-fields aimed at facilitating the appropriate usage of the Instance record. To start with, an Instance may hold an unlimited number of informative fields, however, it is reasonable to assume that it can (and should) be acknowledged by a short display of only some certain representative "leading fields". For instance, in most cases at which a person would have to be addressed, it is reasonable that a display of his first and family names will suffice (no need to specify his Civil ID, gender, age etc.). Accordingly, the Entity field is coupled with the meta-field:

LeadOrd: An integer value that implies a status of a 'leader' field and indicates its order of appearance in short displays of the Entity record. Subsequently, these fields are made available for instance filtering and sorting on the input/output automatic forms and/or reports. A null value will indicate that such a field should not appear in any short display of the record.

The other control characteristics are:

FieldType: An indication of the type of data that PerDB interpreter should expect in the instance field. The indication is actually a pointer onto the branch "Field" of the organization nomenclature tree (parallel to the DataType field of a perceptual tree).

FieldSrc: The address pointing at. The value for this meta-field is expected to be drawn in two stages. First, to choose the relevant table (i.e., Tree, Entity or Collection) from a ComboTree-Box, in which the appropriate branches of either the Tree List or the Entity Heritage Tree will be depicted. Then, the instance will be chosen from the display of the table pointed at.

SubSrc: An auxiliary field that is meant to extract for display only appropriate branches of the FieldSrc selected Tree which will appear thereafter in the FieldSrc ComboTree-Box.

Mandatory: A Boolean indicator for the absolute demand (of the IT) for an updating value once the Instance is processed by the Automatic Input Form.

The third group consists of a single parameter, that is the complementary compartmentalization characteristic (i.e., orthogonal to the Administrative SubOrg) is specified in the field

DataSec: An indicator to the nature of the data expected in the column.
5 Independently, the meta-column DataSec serves to form the field lines (i.e., "information sentence") in the Automatic Input Forms and/or Output Reports. That means that the automatic input/output devices display sequential lines each comprised of a series of instance fields that are characterized by the same DataSec value. The order of lines is determined by the branching order of the DataSec values on the Data Section Tree
10 (T006).

The forth group of the meta-column fields aimed at enabling PerDB to perform adequate data manipulation at various phases of operation. This group comprises the following fields:

ValueLimits: A series of expressions that constitute information used by
15 PerDB for data verification according to predetermined string structures and/or quantity values.

DefaultValue A predetermined string or value that will be put into the referred field whenever a new instance shall be initiated.

OnChange: Macro syntax to be activated upon changing the filed value
20 while operating on the Automatic Input Form, prior to loading the changed values onto the database.

OnUpdate: Macro syntax aimed at manipulating data content of the field referred and/or data of remote fields upon updating the referred filed value.

Properties: A vector of additional properties that may be associated
25 with any field and that are interpreted by the PerDB at run time of the automatic device or within database transactions.

A "Collection" is a device whose records, (i.e., Joints), match a single record of a Tree, an Entity or another Collection (designated "Source") with either a series of values
30 or a series of other records (designated "Target").

Collections may be employed for multi-multi relationships between Instances. Hence, Collections becomes a suitable dispatching recorder, or a highly efficient bi-directional manning reporter.

To meet the Collection task, it is constructed of two administrative and two
5 functional fixed columns. The administrative fields are technically identical to their parallel Entity ones:

ID: The Joint identifying field

Dendrites: The reciprocal pointer vector that records all records that point at the joint excluding the source instance.

10 The fixed columns are:

Sid: A pointer at the source instance. That is, the instance that initiated the connection to the series of targets.

Tid: This field can be used as either a single value, or a pointer at another single record.

15

PerDB handling of the Collection Joints require the meta-characteristics as defined in the "Collection Meta Configuration" L014 list of fields:

CollectionCN: The Collection's Common Name used by PerDB consists of a prefix of the letter "C" followed by the numerator of the table (i.e., Tree, Entity or
20 Collection) whose record (i.e., Tree-Node, Instance or Joint) is the source of the coupling indicated in the herein references Collection's Joint, and followed by a "_i" numerator in accordance with the "Collection_i" field of the "Entity meta Configuration" L012 (see FIGs 1 and 3).

SourceType: An indication to the nature of the Sid field content. It is a
25 pointer to the "Field" branch of the organizational nomenclature tree (T004).

SourceID: An indicator for source table pointing at either the Tree of Trees T000, Entity Heritance Tree T001, or the Collection Meta Configuration L014 ID fields in the case that the initiating record is either a Tree-Node, an Instance or a Joint, respectively.

SourceCN An pointer at the common name of the field that particularly initiated the herein Joint. This indicator is relevant only in the case that the Joint is initiated by an Instance, which may include several fields that initiate a variety of independent Collection Joints.

- 5 **TargetType:** An indication to the nature of the Tid field content. It is a pointer to the "Field" branch of the organizational nomenclature tree (T004).

Target: If the TargetType contains a pointer indicator, this meta-field will hold the address of the record pointed at. If the TargetType contains the "Quantity" indicator, this meta-field will hold the dimensions coupled to the particular parameter.

10

PerDB interpret data elements (e.g., Tree-Node's Data Column, Entity's data Columns and Collection's Tid Column) as follows:

Quantitative elements:

- 15 **Scalar:** A number, integer or real value (such as 2, -1.5, 3.43, etc.), or a logical expression (such as >5, >=7, etc.), that yields a Boolean value of true or false (e.g. 1 or 0 respectively).

- 20 **Quantity:** A scalar followed by a unit vector formulation (*unit/unit...). Units are defined by the "Dimension" branch in Org Nomenclature Tree T004, for example 1200*199/437/968 stands for (in the Tel Hai Academic College's Nomenclature Tree/dimension branch) 1200 New Israeli Shekels (Israel's currency) per semester per student.

- 25 **Field content:** An arithmetic value that is called by the syntax [TableCN].[FieldCN], where [TableCN] may refer to an Entity or refer to an interface (as in the Entity Inheritance Tree [T001].[data]). An arithmetic field is restricted (as determined by its meta-definition) to [FiledType] = integer, real, quantity or Boolean value.

The [TableCN] is mandatory for forms comprised of more than one entity record, which are conjugated by inheritance. If the [TableCN] has left out a default [TableCN] (i.e. the table of the current field considered) is added (e.g. [CN_0], [E026].[CN_0]).

Dynamic association elements – pointers of data fields on records:

The ability to point on a record is restricted to the following fields:

- 5 <tree node>.[data], <entity>.<data fields>, <collections>.[Tid], and is regulated
by one of the following Pointer Terms (taken from the Org Nomenclature Tree T004):
Node Pointer, Instance Pointer, Joint Pointer.

A pointer syntax is in the form of: [TableCN].[ID].<extension>.

- 10 The Instance Pointer (the extension is ineffective): This pointer retrieves leading
data fields by their LeadOrd meta-values. These fields are readily available for sorting,
filtering and similar functions, and also used for triggering full instance expansion.

The Joint Pointer (the extension is ineffective): This pointer retrieves the Tid field
content, be it a self-value, a joint or even a pursuing pointer.

- 15 The Node Pointer (for this pointer the extension is effective): This pointer
retrieves the Label (either Label_H, or Label_E, according to the chosen language of the
user-interface). Data field content is governed by constraints by the rest of the regulatory
technical fields of the Node.

An Extension may take one of the following forms:

.NodeID – Retrieves the Data value of the record identified by NodeID.

- 20 .LatestID – Retrieves the Data value of the record which is the latest update of
the record that holds an identical NodeID value to the record identified by the ID called.

In the absence of an extension the plain ID field is considered as a default.

- 25 PPDO (*i.e.*, People, Perceptions Data and Operations) is a perceptual approach
to information management that makes use of PerDB objects and interpreter engines.
PPDO is a product that starts its life as a minimal series of manufacturer supplied
system-trees and auxiliary tables (sometimes referred to as PerDB-Kernel). System-
trees' Tree-Nodes are tagged "Protected", hence, remain strictly unchanged for the
lifetime of the system and thereby continuously support proper PerDB functioning.

Authorized IT personnel are provided with full capacity to specifically customize their application. They are free to point at the system Tree-Nodes and are encouraged to expand the same system-trees by additional branching, as well as constructing new ones by the organization needs.

5 The PerDB-Kernel comprises the following objects (see FIG 1):

Tree of Trees T000: A perceptual tree, whose nodes depict the hierarchical arrangement of all PPDO trees. Tree of Trees table is the tool by which authorized IT personnel can initiate new trees (*i.e.*, by creating child nodes), then move onto the tree referred to for editing or deletion according to their privileges.

10 Entity Heritage Tree T001: A tree that depicts the heritage relations between Entity tables. It serves also as the first phase in the mechanism of initiating Entity tables.

Input Procedures' Tree T002: A tree that organizes Automatic Input Interface forms in a logical (*i.e.*, a perceptual) order so as to permit laymen users to easily retrieve a desired input form, to activate its associated procedures (*e.g.*: macros) that facilitate
15 inputting data in format of information sentences determined automatically according to the Data Sections Tree (T006) hierarchy.

Output Procedures' Tree T003: A tree with parallel features to those ascribed to the Input Procedures' Tree.

Org Nomenclature Tree T004: A minimal series of terms necessary for
20 proper PerDB interpretation of PPDO Itemization. Hence, prior to the submission of this tree to PPDO customization (*i.e.*, PPDO-Core), it is filled with certain protected branches:

(A) Generic user groups

(B) Privilege level of data manipulation

25 (*i.e.*, hidden, read, execute, write, change, delete).

(C) Statuses

(D) Dimensions

(E) System-Definitions.

IT personnel may clearly use these nodes but never edit or delete them. On the other hand, IT personnel are free to add further specific generations to the same branches to meet their needs.

5 Data Section Tree T005: A tree that compartmentalizes the various definitions of the nature of the organizational data, in the form of predetermined sections. Final nodes in the Data Section Tree define a single elementary unit for security and privileges handling, and are also used for automatic division of input forms into paragraphs comprised of information sentences (series of fields sharing a common meaning and/or nature).

10 Fields are tagged with DataSec value. Grouping is done at run time. The DataSec node in the tree has no knowledge of its consumers.

Manual/Help Navigator T010: An optional, yet highly recommended tree for it offers explanation and help to programmers and users alike. Tree-Nodes point at relevant instances of Manual/Help Item E021.

15 The graphic PPDO editor points to two entities: E012 – Entity Meta editor (for draft editing Meta Entities, this is a programmers or IT personnel tool), and E013 - Field Meta editor (for draft editing fields, again this is a programmers tool, for setting up an ERP application, and not to be used by the end user).

20 At the end of an editing sequence, both these Entities are translated by PerDB (e.g., macros) to three control lists: Entity Meta List L012, which holds the Entities' meta parameters, Field Meta List L013, which holds the Fields' meta parameters and Collection Meta List L014, which holds the collections meta parameters that are drawn from the collection originating Entity Fields' meta characteristics. Table Nominator L011, is an auxiliary list of numbers used to assure exclusive assignment of table indices to
25 Trees, Entities and Interfaces.

The PerDB-Kernel is in fact a programmers (or IT personnel) tool, for this is the infrastructure of the ERP application in accordance with a preferred embodiment of the present invention.

30 A practical ERP application in accordance with a preferred embodiment of the present invention suitable for adaptation for any organization requires an expansion of

PerDB-Kernel. Any organizations who exercises information compartmentalization, needs to inform PerDB on its personnel, suppliers and customers accessibility privileges. Accordingly, PerDB-Kernel is supplemented with a minimal set of tables (combined they are sometimes referred to as PPDO-Core, see FIG 6).

- 5 In addition to the PerDB-Kernel (see FIG 1) the PPDO application requires basic information on the organization manning definitions and the compartmentalization parameters of the organization's registered personnel, suppliers and customers, Accordingly PPDO-Core comprises the following tables (see FIG 6 central box)

- 10 Org manning Tree T006: The hierarchical presentation of the manning of the organization echelons. One should keep in mind that PerDB, and consequently PPDO, are information managing applications. Thus, the Tree-Node of the Org Manning Tree should not report merely the echelon's judicial state, but rather depict all types of its manning (i.e., personnel, suppliers and customers). For the sake of suitable information compartmentalization, special emphasis should be made with regard to the broadening and narrowing (i.e., generalization vs. specialization) of the Tree-Node hierarchy.

15 Org Job Tree T021: The categorization of all types of jobs within the organization. The categorization resolution should reflect the information gathered on the appointees, and the information that these appointees are expected to have accessibility to. Each T021 Tree-Node points at a certain Job Definition Instance:

- 20 Job Definitions E022: The detailed description of a job in terms of responsibilities and interactions with other organization appointees as well as out of the organization suppliers and customers. Each job definition Instance is connected via Collection C022_0 to a series of privilege definitions:

- 25 Job Privileges E023: A pointer at a Data Section Tree (T005) branch coupled with a pointer at an Action Privilege branch of the Org Nomenclature Tree (T004). Each such couple defines two (out of three) compartmentalization flags. Accumulative privileges of a given job are grouped with the aid of collection C022_0 in a one-to-many reference. The third compartmentalization (i.e., SubOrg) flag is attributed to an appointee as by its organizational assignment.

- 30 Person Card E025: An array of data fields that identify the persons within the organization. Usually it comprises the person's name, civil ID number,

organization's ID number, and sometimes, the computer account User-Name and current password. A Person Card is intermittently coupled with:

5 Personal Item Card E026: An expansion table which includes complementary details of the person as by the organization policy. Except for pointers on all appointments of the referred person are absolutely necessary for proper PPDO customization, in particular, for data compartmentalization. A person may hold one or many appointments in the organization. Accordingly, the Personal Items Card is connected via Collection C026_0 to a series of:

10 Appointment E027: An array of pointers that define first a job (i.e. pointer on T021), which is automatically linked to the appropriate Job Definition and series of DataSec times Action Privilege parameters. Second, it specifies the affiliation (i.e., a pointer on a single T006 Tree-Node) in context of which the job is defined, thus, indicating who hired that person (i.e. who is/are his superior/s). Finally, it defines the job organizational jurisdiction (i.e., a series of pointers on T006 manning group braches) in terms of who that person is responsible for (i.e., who is/are his "organizational customers")..

20 Unlike the PerDB-Kernel tables, the PPDO-Core complementary tables are provided empty of content. Since the latter is a pure reflection of the organization's specific way of operation. Nonetheless, their crucial task in putting PerDB to work (i.e., PPDO customization) requires the attention of IT personnel at the very early stages of their system construction.

25 The following explanation goes into explaining the present invention in greater detail, by way of example, based on a working basic framework that was built by the inventors of the present invention to serve as the administrative management system for the Tel-Hai College (in Northern Galilee, Israel). This application was built based on Microsoft's MS-SQL platform, but this is of course not mandatory, as the present invention may be implemented on other T-SQL supporting platforms too.

30 FIG.7 illustrates a module of an ERP application in accordance with a preferred embodiment of the present invention that was designed for The Tel-Hai Academic College Marketing department.

 This module aims at assisting The Marketing Department personnel process

Student Application that enter the system either by a web based self application, offering applicants the chance to apply to the college via the internet, or by an agent who types the data in while having verbal communication with the applicant.

On the first route, input is made into Self Application Entity E179 using an AST
5 from accessible on the Tel Hai Academic College web site. A process (macro) is initiated for transferring the data into an Applicant Card E175, so that Tel Hai personnel could inspect and approve the information from an adequate Tree-Node of the Input Procedures' tree T002. The Applicant Card E175 points at the status branch of the Org Nomenclature Tree T004/status branch to extract an Applicant Status value. An
10 Applicant Card persists for the system life time, hence, it is connected to many Application Card E176 via collection C175_0, which represents a single dialogue (session) between the applicant and the Tel Hai Marketing Department, irrelevant of the dialogue mode (i.e., internet, phone call or letter). The Application Card E177 points to three branches of the Org Nomenclature Tree T004 to extract an Application Status
15 value, an indication of who recommended the college to the applicant, and an indication of mode of communication employed by the Applicant to execute the application. During an application session a series of Applicant's subjects of interest could be uploaded to a Subject of Interest entity (E177) with the aid of Collection C176_0. Subjects of interest are specified by pointing to three branches: one extracts a general discipline of interest
20 from Tel-Hai's Nomenclature Tree T004; the second, extracts a department of interest from Tel-Hai's Manning tree T006/student branches, and the last extracts a specific program of interest from Tel-Hai's Teaching Program Tree T031. The Application Card is also connected, via collection C176_1 to Marketing Response Entity E178, itself connected to Collection C178_0 whose Tid column holds pointers to Tel Hai's
25 Nomenclature Tree T004/Public Relations (PR) materials branches.

The above framework determines the pattern of data collection. It is an exclusive scheme that is unequivocally interpreted by PerDB in offering the Marketing Department of the college personnel Automatic Input Forms while holding the dialogues with the applicants. Notice that the Marketing Department personnel may enter the scheme at
30 any relevant point, be it with registering a new applicant, or be it an opening of a successive session with an already registered applicant, or even a general entry to subject of interest or list of responses carried out by the Marketing Department

Personnel. In any such entry the neural-network properties of the instances (i.e., "Dendrites" values) would automatically trigger the opening of relevant upstream information (e.g., for an entry of either Subject of Interest or Marketing response indications to the relevant Application and Applicant Cards will be displayed).

- 5 The example shown in FIG. 7 demonstrates the universal appeal of the EPR application of the present invention, which is in fact a flexible tool that may be adapted to any organization and serve it properly.

- 10 It should be clear that the description of the embodiments and attached Figures set forth in this specification serves only for a better understanding of the invention, without limiting its scope as covered by the following Claims.

 It should also be clear that a person skilled in the art, after reading the present specification could make adjustments or amendments to the attached figures and above described embodiments that would still be covered by the following Claims.

C L A I M S

1. A method for a universal Enterprise Resource Planning (ERP) information management suitable for any organization, the information separated into raw data and
5 at least one of a plurality of subjective perceptions attached to it, the method comprising:
organizing information into tables of three types:
 - entity table for raw data,
 - tree table for perception ,
 - collection table for many-to-many associations,
- 10 manipulating the information according to a set of predetermined parameters that constitutes meta characteristics of the information, using a set of macros to perform operations.
2. The method of claim 1, wherein the tables comprise a database.
3. The method of claim 1, wherein entity table comprise administrative fields
15 which record inheritance.
4. The method of claim 1, wherein entity table comprise administrative fields which record reciprocal pointing relations.
5. The method of claim 1, wherein entity table comprise administrative fields which record interface implementations.
- 20 6. The method of claim 1, wherein entity table comprise administrative fields which record affiliation to an echelon of the organization.
7. The method of claim 1, wherein entity table comprise informative fields for raw data.
8. The method of claim 1, wherein tree-table comprise records defining tree-
25 nodes.
9. The method of claim 1, wherein tree-table comprise multi-lingual fields.
10. The method of claim 12, wherein one of the multi-lingual fields relates to a

nomendature of the organization.

11. The method of claim 1, wherein tree-nodes comprise fields relating to tracking of evolution of the tree-nodes.
12. The method of claim 1, wherein tree-nodes comprise fields relating to compartmentalization.
13. The method of claim 1, wherein tree-nodes comprise fields relating to one or many entity or collection records.
14. The method of claim 1, wherein tree-nodes comprise fields relating to tree-node to tree-node association and/or disassociation
15. The method of claim 1, wherein tree-nodes comprise fields relating to a cascade of tree-nodes and thereafter relating to their related tree-nodes and/or entity records and/or collection records.
16. The method of claim 1, wherein collection table comprises administrative fields holding one-to-one relations between records.
17. The method of claim 1, wherein collection table comprise administrative fields holding one-to one relation between a record and a value.
18. The method of claim 1, wherein collection table comprise administrative fields which record reciprocal pointing relations.
19. The method of claim 1, wherein the set of predetermined parameters comprises one or more tables.
20. The method of claim 1, wherein the set of predetermined parameters is modifiable.
21. The method of claim 1, wherein the set of predetermined parameters that constitutes meta characteristics is stored in a kernel.
22. The method of claim 1, wherein the set of predetermined parameters that constitutes meta characteristics of entities.
23. The method of claim 1, wherein the set of predetermined parameters that constitutes meta characteristics of fields.

24. The method of claim 1, wherein the set of predetermined parameters that constitutes meta characteristics of collections.
25. The method of claim 1, wherein the operations also include functions.
26. The method of claim 1, wherein the operations include automatic operators that create or modify tables and further meta characteristics using a predetermined editor.
27. The method of claim 1, wherein the operations dynamically create automatic input forms and/or output reports in accordance with the meta characteristics.
28. The method of claim 27, wherein the automatic input forms include a contracting/expanding tree.
29. The method of claim 1, wherein the operations dynamically guard validity of data in accordance with the meta-characteristics.

Fig. 1

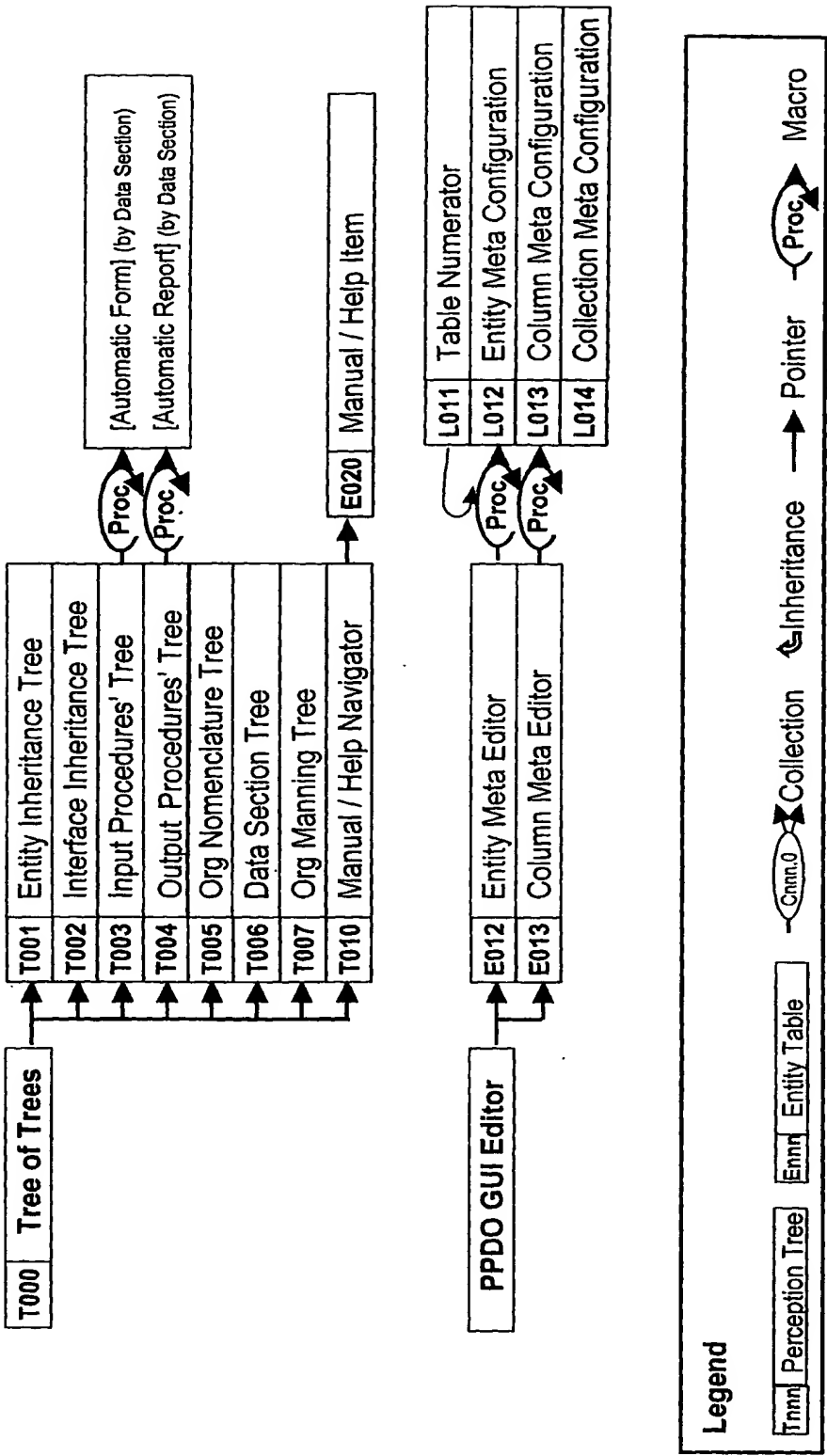


Fig. 2

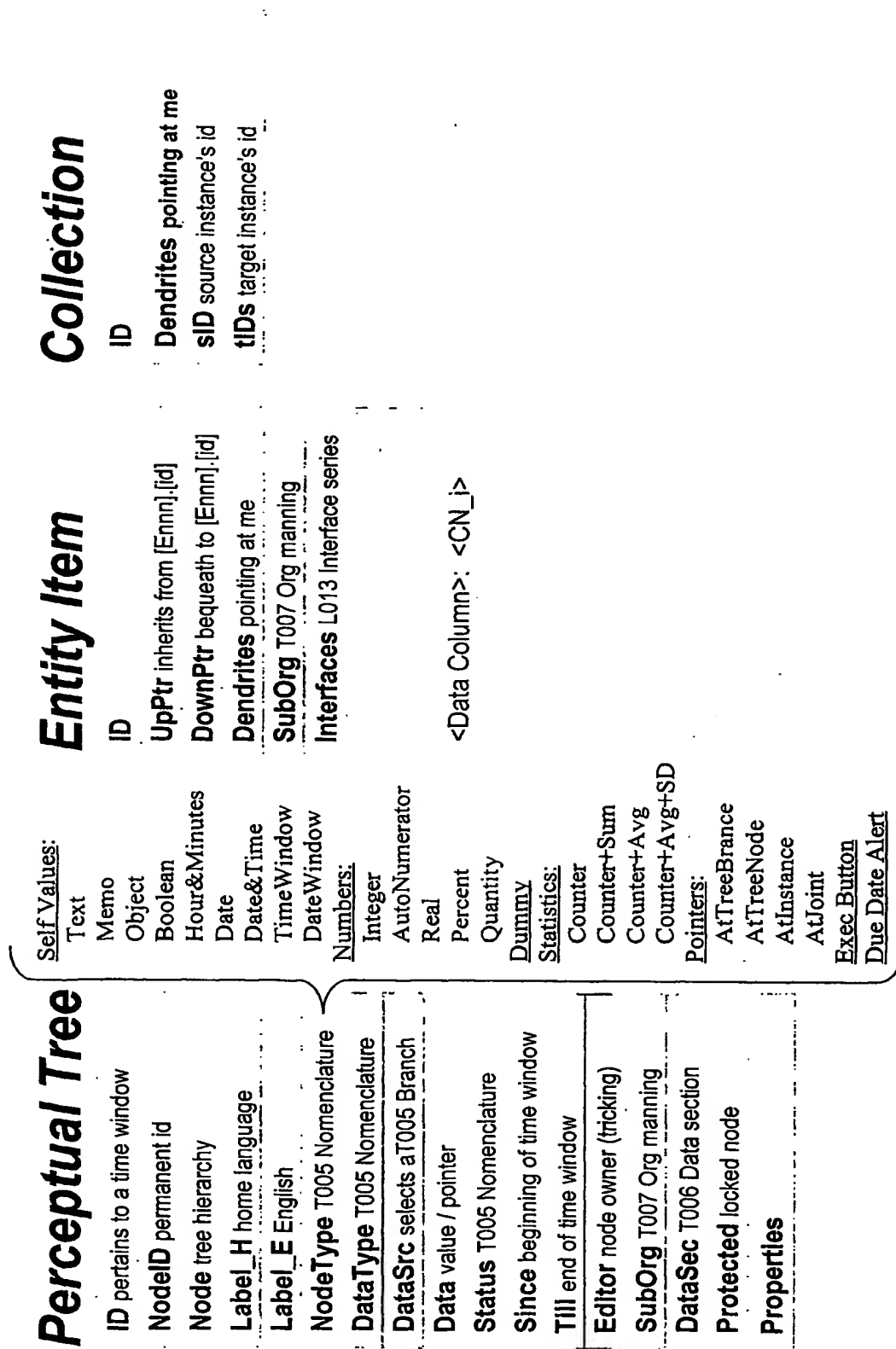


Fig. 3

Perceptual Tree

None

Entity / Interface

listed on L012

Entity_CN from L011
Label_H home language
Label_E English
DataSec T006 Data section
Abstract Boolean
Final Boolean
CN_J Next column common name
Collection_J Next Collection index
DenyDendrites Boolean
Interfaces vector of [Innn]
SubOrgFieldCN SubOrg source
SubOrgOverride Default by T007

Collection

listed on L014

ID
CN Collection table name]
Source Field Type by T005
SourceID Source record id
SourceCN Source field name
Target Target Type by T005
TargetID Target record id

Fig. 5

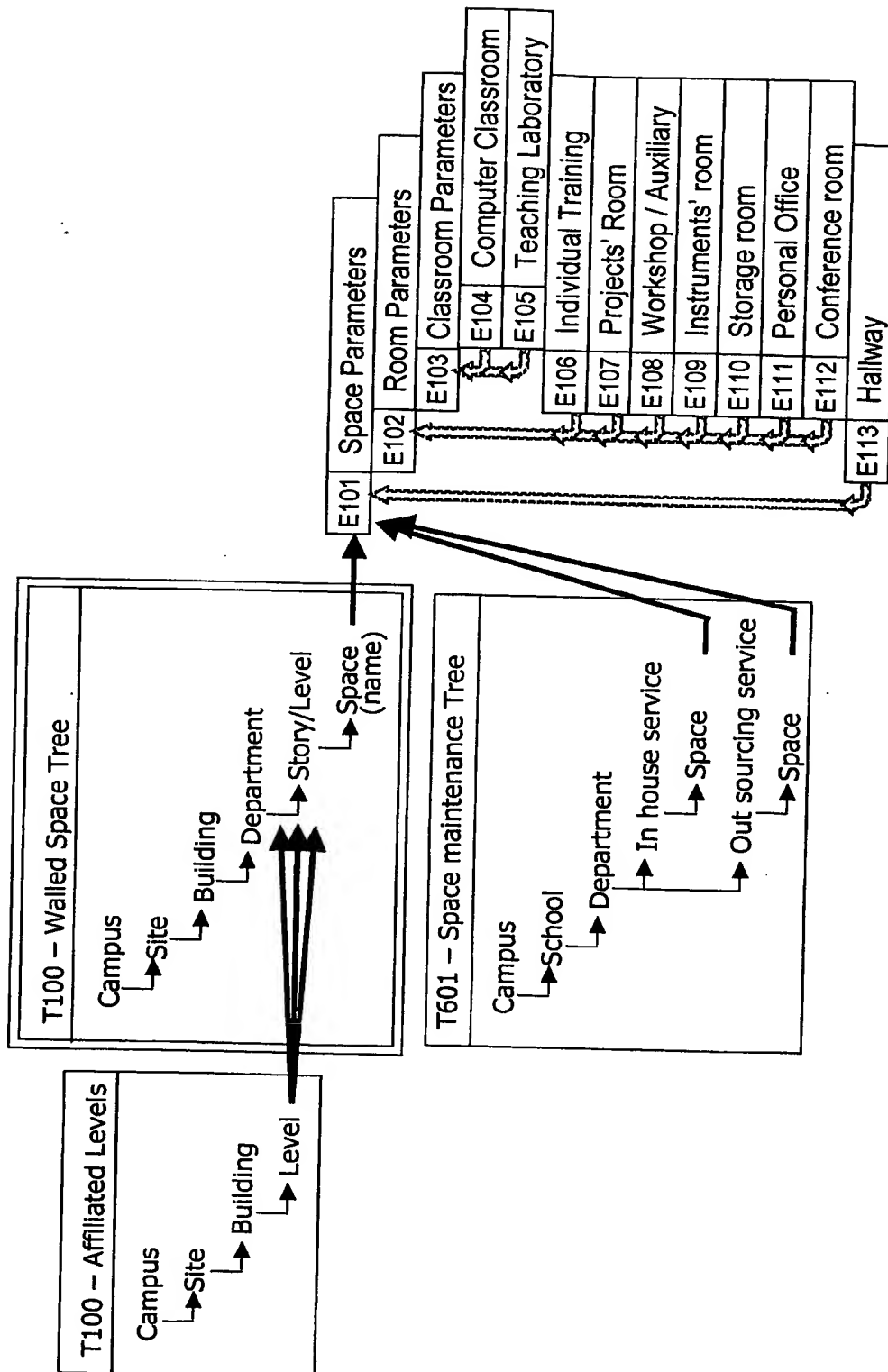


Fig. 6

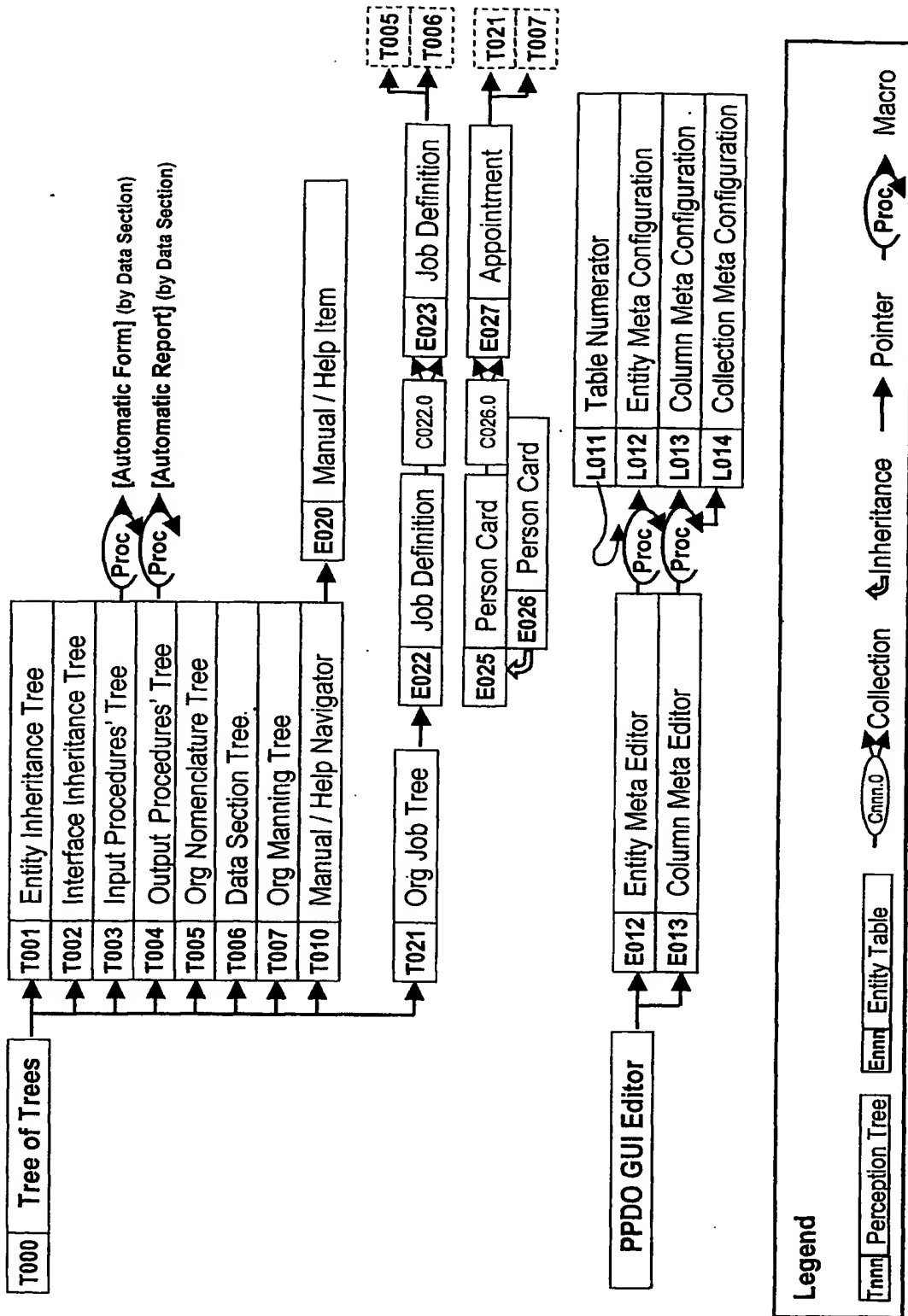


Fig. 7

